Information Retrieval Models for Contact Recommendation in Social Networks

Javier Sanz-Cruzado[(✉)] and Pablo Castells[(✉)]

Universidad Autónoma de Madrid, Escuela Politécnica Superior, Madrid, Spain
{javier.sanz-cruzado,pablo.castells}@uam.es

Abstract. The fast growth and development of online social networks has posed new challenges for information retrieval and, as a particular case, recommender systems. A particularly compelling problem in this context is recommending network edges, that is, automatically predicting people that a given user may wish or benefit from connecting to in the network. This task has interesting particularities compared to more traditional recommendation domains, a salient one being that recommended items belong to the same space as the users they are recommended to. In this paper, we investigate the connection between the contact recommendation and the text retrieval tasks. Specifically, we research the adaptation of IR models for recommending contacts in social networks. We report experiments over data downloaded from Twitter where we observe that IR models, particularly BM25, are competitive compared to state-of-the art contact recommendation methods. We further find that IR models have additional advantages in computational efficiency, and allow for fast incremental updates of recommendations as the network grows.

Keywords: Social networks · Contact recommendation ·
Text information retrieval

1 Introduction

The creation of online social network applications such as Twitter, Facebook and LinkedIn, and their subsequent expansion along the 2,000s has given rise to new perspectives and challenges in the information retrieval (IR) field, and, as a particular case, recommender systems. One of the most compelling problems in this area is recommending people with whom users might want to engage in an online network. The social nature of these networks, and the massive amount of users accessing them every day has raised the interest for contact recommendation of both industry [10, 11] and several research communities [4, 12, 13]. The most prominent social platforms offer user recommendation services since the end of the past decade, with systems such as 'Who-to-follow' on Twitter [10, 11] or 'People you may know' on Facebook and LinkedIn.

Contact recommendation represents a very particular perspective of the recommendation task. On the one hand, the recommendation domain lays connections to social network analysis and network science, with rich potential implications [12, 28]. On the other, while in most domains users and items are different objects, this one has

the peculiar and interesting characteristic that users and items are the same set. These particularities have motivated the creation of a wide variety of people recommendation algorithms from diverse fields, such as network science [18, 19], machine learning [14], recommender systems [13] or, to a lesser extent, information retrieval [13].

In our present work we focus on this last line of research: we investigate the relation between contact recommendation in social networks, and text retrieval. For this purpose, we establish associations between the fundamental elements involved both tasks, in order to adapt classic IR models to the task of suggesting people in a social network. We explore the adaptation of well-known models: the vector space model [26] (VSM), BM25 [25] and query likelihood [22]. We empirically compare the effectiveness of the resulting algorithms to state-of-the-art contact recommendation methods over data samples extracted from Twitter, and we find the adapted IR models, particularly BM25, to be competitive with the best alternatives. Moreover, we find important additional advantages in terms of computational efficiency, both in producing recommendations from scratch, and in incrementally updating them as the network grows with new links and users.

2   Related Work

In the context of online social networks, contact recommendation aims at identifying people in a social network that a given user would benefit from relating to [30]. The problem is in many aspects equivalent to the link prediction task [18, 19], which aims to identify unobserved links that exist or will form in the future in a real network. Link prediction and recommendation is an established topic at the confluence of social network analysis and recommender systems for which many methods have been proposed in the literature, based on the network topology [18], random walks across the network graph [4, 10], or user-generated content [13].

In this paper, we investigate the adaptation of classic text IR models to the contact recommendation task. The connections between recommendation and text IR date back to the earliest recommender systems and their relation to the information filtering task [6]. Even though most of this connection has focused on content-based methods [2], it has also developed into collaborative filtering algorithms [7, 31, 32].

A particularly representative and relevant approach for our present work was developed by Bellogín et al. [7], allowing the adaptation of any IR term weighting scheme to create a collaborative filtering algorithm. To this end, the approach represents users and items in a common space, where users are the equivalent of queries, and items play the role of the documents to be retrieved. Our work pursues a similar goal, but taking a step further: if Bellogín et al. folded three spaces (terms, documents, queries) into two (users, items), we fold them into just one, as we shall explain.

Some authors have likewise connected IR techniques to the specific task of recommending users in social networks. For example, some link prediction approaches, such as the ones based on the Jaccard index [16, 18, 27] have their roots in IR. More recently, Hannon et al. [13] adapted the vector-space model [26] to recommend users on

Twitter, based on both content-based and collaborative filtering algorithms. Our work seeks to extend, generalize and systematize this point of view to adapt any state-of-the-art IR model to contact recommendation.

3   Preliminaries

We start by formally stating the contact recommendation task, and introducing the notation we shall use in our formulation. We can represent the structure of a social network as a graph $\mathcal{G} = \langle \mathcal{U}, E \rangle$, where $\mathcal{U}$ is the set of network users, and $E \in \mathcal{U}_*^2$ is the set of relations between users (friendship, interactions, whatever the network is representing), where $\mathcal{U}_*^2 = \{(u, v) \in \mathcal{U}^2 | u \neq v\}$ is the set of pairs formed by different users.

For each user $u \in \mathcal{U}$, we denote her neighborhood as $\Gamma(u)$ (the set of users that u has established relations with). In directed networks, three different neighborhoods can be considered: the incoming neighborhood $\Gamma_{\text{in}}(u)$ (users who create links towards u), the outgoing neighborhood $\Gamma_{\text{out}}(u)$ (users towards whom u creates links), and the union of both neighborhoods $\Gamma_{\text{und}}(u)$. In weighted graphs we have additionally a weight function $w : \mathcal{U}_*^2 \to \mathbb{R}$, which returns the weight of an edge if $(u, v) \in E$, and 0 otherwise. In unweighted graphs, we can consider that $w(u, v) = 1$ if the link exists, and 0 otherwise.

Now given a target user u, the contact recommendation task consists in finding a subset of users $\tilde{\Gamma}_{\text{out}}(u) \subset \mathcal{U} \setminus \Gamma_{\text{out}}(u)$ towards whom u has no links but who might be of interest for her. We address the recommendation task as a ranking problem, in which we find a fixed number of users $n = \left| \tilde{\Gamma}_{\text{out}}(u) \right|$ sorted by decreasing value of a ranking function $f_u : \mathcal{U} \setminus \Gamma_{\text{out}}(u) \to \mathbb{R}$.

4   IR Model Adaptation Framework for Contact Recommendation

Even though recommendation and text retrieval have been traditionally addressed as separate problems, it is possible to establish analogies and equivalences between both tasks. Recommender systems are indeed often described as retrieval systems where the query is absent, and records of user activity are available instead [7], and the approaches we develop follow this perspective.

4.1   Task Unification

In order to adapt text IR models to the recommendation task, we need to establish equivalences between the elements in the contact recommendation task (users and interactions between them) and the spaces involved in text search (queries, documents and terms). In previous adaptations of IR models for recommendation, the three IR spaces commonly folded into two: the set of users and the set of items [7]. However, when we seek to recommend people in social networks, the latter two spaces are the

same. Therefore, to adapt the IR models to our task, we fold the three IR spaces into a single dimension: the set of users in the social network, playing the three different roles, as we illustrate in Fig. 1. We explain next in more detail how we carry this mapping through.
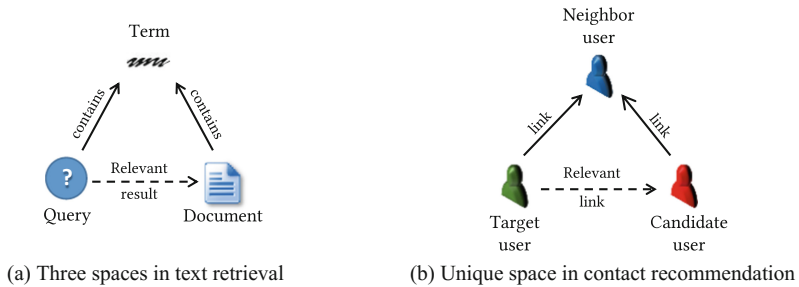

Fig. 1. Text IR elements (a) vs. contact recommendation elements (b).
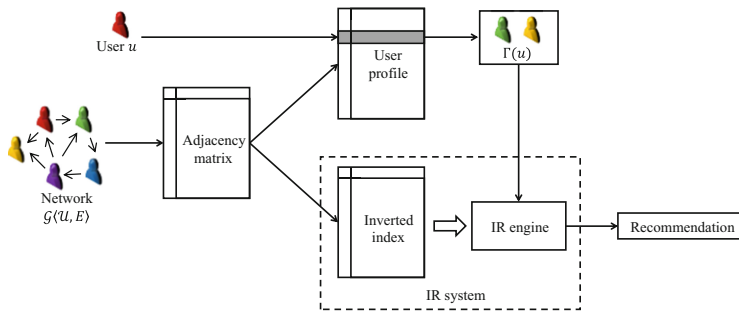

Fig. 2. Adaptation of IR models to recommend users in social networks.

First, the natural equivalent of documents in the search space are candidate users (to be recommended as contacts), as they play the same role: they are the elements to be retrieved in order to fulfil a user need. The need is explicit in the search task, expressed by a query; and it is implicit in contact recommendation: the need for creating new bonds. This social need is to be predicted based on records of past user activity, which therefore play an equivalent role to the query keywords in text IR. In a social network, past user activity is encoded in existing links to and/or from the target user.

Finally, we need an equivalent to the term representation of documents. In prior adaptations of IR models for recommendation, this was the main difficulty: users and items were different objects, so a representation that suits one might not work for the other [7]. In contact recommendation this becomes in fact easier: users and items are the

same thing, so any term representation for target users is automatically valid for the "items" (the candidate users). The possibilities for defining an equivalent to terms are manifold, and result in very different algorithms. For instance, we can define content-based recommendation methods by using texts associated to users, such as messages or documents posted or liked by the users [13]. On the other hand, if we take users as the term space, and we equate the term-document relationship to interactions between users, we obtain collaborative filtering algorithms. We shall focus on the latter approach in this paper.

Figure 2 illustrates the collaborative filtering adaptation approach. A social network is encoded as a weighted adjacency matrix A, where $A_{uv} = w(u, v)$. Using link data, we build two elements: on one hand, an inverted index that allows for fast retrieval of candidate users and, on the other, a structure that provides direct access to the neighborhood of the target users, i.e. the query term representation. The inverted index uses network users as keys (playing the role of terms), and postings lists store the set of candidate users to whose neighborhood representation (as "documents") the "key" users belong to.

Using this index and the "query" structure, any text IR engine can be used as a contact recommendation algorithm. Additional details and options remain open how-ever when developing a specific instance of this framework in full detail, as we will describe in the following sections. An important one concerns the direction of social links in the reinterpretation of IR models, to which we shall pay specific attention.

4.2 Neighborhood Orientation

In directed social networks such as Twitter or Instagram, three definitions of user neighborhood can be considered: the incoming neighborhood $\Gamma_{\text{in}}(u)$, the outgoing neighborhood $\Gamma_{\text{out}}(u)$ and the union of both, $\Gamma_{\text{und}}(u) = \Gamma_{\text{in}}(u) \cup \Gamma_{\text{out}}(u)$. Any of the three options is valid in our adaptation of IR models. Since the inverted index and user profiles are created independently, it is even possible to take a different choice for target and candidate users: since we still use the same elements to represent (the equivalent of) both queries and documents, it is possible to work just smoothly with different neighborhood orientation choices for targets and candidates.

Identifying which neighborhood best characterizes the candidate and target users in the social network is an interesting problem by itself [13]. It concerns many state-of-the-art contact recommendation algorithms –besides IR adaptations– such as Adamic-Adar [1] or Jaccard similarity [18, 27] which use neighborhoods in their ranking functions. We shall therefore explore this issue in our experiments in Sect. 6.

5 Adaptation of Specific IR Models

As an example of the general unification framework, we now show in some detail the adaptation of two particular IR models: BIR and BM25 [25]. In the formulations in this section, we shall denote the neighborhood representation of the target user as $\Gamma^q(u)$, and the neighborhood representation of the candidate users as $\Gamma^d(v)$.

5.1   Binary Independence Retrieval

The model known as BIR (binary independence retrieval) [25] is the simplest representative of IR models building on the probability ranking principle [24]. Under the assumption that term occurrence follows a (multiple) Bernoulli distribution, this model estimates the probability of relevance of a document d for a query q as:

$$P(r|d, q) \propto \sum_{t \in d \cap q} RSJ(t) \tag{1}$$

where r denotes the event that the document is relevant, and RSJ is the Robertson-Spärck-Jones formula [25], which is defined as:

$$RSJ(t) = \log \frac{|R_t|(|D| - |D_t| - |R| - |R_t|)}{(|R| - |R_t|)(|D_t| - |R_t|)} \tag{2}$$

In the above equation R is the set of relevant documents for the query, R_t is the set of relevant documents containing the term t, D is the document collection, and D_t is the set of documents containing t. Since the set R of relevant documents is not known, the following approximation can be taken, considering that typically only a tiny fraction of documents are relevant:

$$RSJ(t) = \log \frac{|D| - |D_t| + 0.5}{|D_t| + 0.5} \tag{3}$$

As described in Sect. 4, to adapt this model for contact recommendation, we equate queries and documents to target and candidate users respectively, and the term-document relationship to social network edges. Under this equivalence, $|D|$ is the number of users in the network, and $|D_t|$ is the number of users that t is a neighbor of (i.e. her neighbor size in the transposed network). Denoting inverse neighborhoods as $\Gamma_{inv}^d(t)$, the adapted BIR equation becomes:

$$f_u(v) = \sum_{t \in \Gamma^q(u) \cap \Gamma^d(v)} RSJ(w) = \sum_{t \in \Gamma^q(u) \cap \Gamma^d(v)} \log \frac{|\mathcal{U}| - |\Gamma_{inv}^d(t)| + 0.5}{|\Gamma_{inv}^d(t)| + 0.5} \tag{4}$$

5.2   BM25

BM25 is one of the best-known and most effective probabilistic IR models [25]. It starts from similar principles as BIR, but modeling term occurrence in documents as a Poisson instead of a Bernoulli distribution. Its ranking function is defined as:

$$P(r|d, q) \propto \sum_{t \in d \cap q} \frac{(k+1)\text{freq}(t, d)}{k(1 - b + b|d|/\text{avg}_{d'}(|d'|)) + \text{freq}(t, d)} RSJ(t) \tag{5}$$

where freq(t, d) denotes the frequency of t in d, $|d|$ is the document length, RSJ(w) is defined in Eq. 3, and $k = [0, \infty)$ and $b \in [0, 1]$ are free parameters controlling the effect of term frequencies and the influence of the document length, respectively.

The text retrieval space can be mapped to a social network just as before, now taking, additionally, edge weights as the equivalent of term frequency. In directed networks, we will need to make a choice between the weight of incoming or outgoing links as the equivalent of frequency. We shall link this decision to the edge orientation selected for candidate users (as pointed out earlier in Sect. 4.2 and beginning of Sect. 5), as follows:

$$\text{freq}(t, v) = w^d(v, t) = \begin{cases} w(t, v) & \text{if } \Gamma^d \equiv \Gamma_{\text{in}} \\ w(v, t) & \text{if } \Gamma^d \equiv \Gamma_{\text{out}} \\ w(v, t) + w(t, v) & \text{otherwise} \end{cases} \quad (6)$$

Finally, document length can be now defined as the sum of edge weights of the candidate user. In unweighted graphs this is simply equivalent to the degree of the node; in directed networks we have again different choices. The BM25 formulation for text retrieval considers different options in defining document length (number of unique terms, sum of frequencies, etc.) [25]. We have found similarly worthwhile to decouple the orientation choice for document length from the one for the term representation of candidate users. We reflect this by defining length as:

$$\text{len}^l(v) = \sum_{t \in \Gamma^l(v)} w^l(v, t) \quad (7)$$

where $\Gamma^l(v)$ represents the candidate's neighborhood in a specific orientation choice for document length. Based on all this, the adaptation of BM25 becomes:

$$f_u(v) = \sum_{t \in \Gamma^q(u) \cap \Gamma^d(v)} \frac{(k+1)w^d(v, t)}{k\big(1 - b + b\, \text{len}^l(v)/\text{avg}_x \text{len}^l(x)\big) + w^d(v, t)} \text{RSJ}(t) \quad (8)$$

5.3   Other IR Models

Analogous adaptations can be defined for virtually any other IR model, such as the vector space model [26] or query likelihood [22], which we summarize in Table 1, including Jelinek-Mercer [17] (QLJM), Dirichlet [20] (QLD), and Laplace smoothing [32] (QLL) for query likelihood, which were adapted in prior work for general recommendation [7, 31, 32] –we now adapt them to the specific contact recommendation task.

Table 1. Adaptation of IR models to contact recommendation.

Model	Ranking function
VSM	$f_u(v) = \displaystyle\sum_{t\in\Gamma^q(u)\cap\Gamma^d(v)} u_t v_t \Big/ \sqrt{\displaystyle\sum_{t\in\Gamma^d(v)} v_t^2}$
	$u_t = \mathrm{tf}-\mathrm{idf}^q(u,t) = w^q(u,t)\log\big(1+|\mathcal{U}|/\big(1+\big|\Gamma_{\mathrm{inv}}^q(t)\big|\big)\big)$
	$v_t = \mathrm{tf}-\mathrm{idf}^d(v,t)$
BIR	$f_u(v) = \displaystyle\sum_{t\in\Gamma^q(u)\cap\Gamma^d(v)} \mathrm{RSJ}(t) \qquad \mathrm{RSJ}(t) = \log\dfrac{|\mathcal{U}|-\big|\Gamma_{\mathrm{inv}}^d(t)\big|-0.5}{\big|\Gamma_{\mathrm{inv}}^d(t)\big|-0.5}$
BM25	$f_u(v) = \displaystyle\sum_{t\in\Gamma^q(u)\cap\Gamma^d(v)} \dfrac{(k+1)w^d(v,t)\cdot\mathrm{RSJ}(t)}{k\big(1-b+b\,\mathrm{len}^l(v)/\mathrm{avg}_x\,\mathrm{len}^l(x)\big)+w^d(v,t)}$
QLJM	$f_u(v) = \displaystyle\sum_{t\in\Gamma^q} w^q(u,t)\log\left((1-\lambda)\dfrac{w^d(v,t)}{\mathrm{len}^d(v)} + \lambda\dfrac{\mathrm{len}_{\mathrm{inv}}^d(t)}{\sum_{x\in\mathcal{U}}\mathrm{len}^d(x)}\right)$
QLD	$f_u(v) = \displaystyle\sum_{t\in\Gamma^q} w^q(u,t)\log\left(\dfrac{w^d(v,t)+\mu\,\mathrm{len}_{\mathrm{inv}}^d(t)/\sum_{x\in\mathcal{U}}\mathrm{len}^d(x)}{\mathrm{len}^d(v)+\mu}\right)$
QLL	$f_u(v) = \displaystyle\sum_{t\in\Gamma^q} w^q(u,t)\log\left(\dfrac{w^d(v,t)+\gamma}{\mathrm{len}^d(v)+\gamma|\mathcal{U}|}\right)$

6 Experiments

In order to analyze the performance of the adaptation of IR methods to contact rec-
ommendation and compare them to baseline alternatives, we conduct several offline
experiments using social network data extracted from Twitter. We describe the
experimental approach, setup and results in the paragraphs that follow.

6.1 Data and Experimental Setup

We run our experiments over dynamic, implicit networks induced by the interactions
between users (i.e. $(u,v)\in E$ if u retweeted, mentioned or replied v). We built two
datasets: one containing all tweets posted by a set of around 10,000 users from June
19th to July 19th 2015, and one containing the last 200 tweets posted by 10,000 users as
of August 2nd 2015. Users are sampled in a snowball graph crawling approach starting
with a single seed user, and taking the interaction tweets (retweets, mentions, replies)
by each user as outgoing network edges to be traversed. User sampling stops when
10,000 users are reached in the traversal; at that point, any outgoing edges from
remaining users in the crawl frontier pointing to sampled users are added to the
network.

For evaluation purposes, we partition the network into a training graph that is
supplied as input to the recommendation algorithms, and a test graph that is held out
from them for evaluation. IR metrics such as precision, recall or nDCG [5] can be
computed on the output of a recommendation algorithm by considering test edges as
binary relevance judgments: a user v is relevant to a user u if –and only if– the edge
(u,v) appears in the test graph. In our experiments we apply a temporal split, which

Table 2. Twitter network dataset details.

Network	Users	Training edges	Test edges
1 Month	9,528	170,425	54,355
200 Tweets	9,985	137,850	21,598

better represents a real setting: the training data includes edges created before a given time point, and the test set includes the links created afterwards. The split point for the "1 month" dataset is July 12th (thus taking three weeks for training and one for test); and in "200 tweets" the split date is July 29th in order to have 80% of edges in the training graph. Edges appearing in both sides of the split are removed from the test network, and the frequency of training interaction between every pair of users is available to the evaluated systems as part of the training information. We show the resulting dataset statistics in Table 2.

Finally, to avoid trivializing the recommendation task, reciprocating links are excluded from both the test network and the systems' output. Given the high reciprocation ratio on Twitter, recommending reciprocal links would be a trivial hard to beat baseline. Moreover, users already notice when someone retweets or mentions them since Twitter sends notifications every time, whereby an additional recommendation would be redundant and would barely add any value.

6.2   Recommendation Algorithms

We assess the IR model adaptations by comparing them to a selection of the most effective and representative algorithms in the link prediction and contact recommendation literature. These include Adamic-Adar [1], most common neighbors (MCN) [18], personalized PageRank [1], and collaborative filtering (item-based and user-based kNN [21], and implicit matrix factorization (iMF) [15], as implemented in the RankSys library [23]). In addition, we implement the Money algorithm [10, 11] developed at Twitter, in which, for simplicity, we include all users in the circle of trust. We also include random and most-popular recommendation as sanity-check baselines.

We optimize all algorithms (edge orientation and parameter settings) by grid search targeting P@10. For those that can take advantage of edge weights (IR models and collaborative filtering algorithms), we select the best option. The resulting optimal settings are detailed in Table 3.

6.3   Experimental Results

We show in Table 4 the results for both datasets. We observe that only four of the algorithms in our comparison achieve good results in both datasets: the implicit matrix factorization approach, BM25 and, to a lesser extent, Adamic-Adar and BIR. Indeed, iMF is the best algorithm in terms of precision and recall for the "1 month" dataset, whereas BM25 achieves the maximum accuracy in terms of P@10 for the "200 tweets" dataset, with a technical tie (non-significant difference) in R@10. For the rest of algorithms, we see three different trends: Jaccard and VSM are far from the best

Table 3. Parameter settings for each algorithm and dataset. We take $\Gamma^q \equiv \Gamma_{und}$ and $\Gamma^d \equiv \Gamma_{in}$ for all algorithms, except $\Gamma^d \equiv \Gamma_{und}$ for VSM on 200 tweets. For BM25 we take $\Gamma^l \equiv \Gamma_{out}$. All algorithms perform best without weights, except BM25 on both datasets, and VSM on 1 month. In Adamic-Adar, Γ^l represents the direction on the selection of common neighbors between the target and candidate users (see [30]).

Algorithm	1 Month	200 Tweets
BM25	$b = 0.1, k = 1$	$b = 0.5, k = 1$
QLD	$\mu = 1000$	$\mu = 1000$
QLJM	$\lambda = 0.1$	$\lambda = 0.1$
QLL	$\gamma = 100$	$\gamma = 100$
Money	Authorities, $\alpha = 0.99$	Authorities, $\alpha = 0.99$
Adamic-Adar	$\Gamma^l \equiv \Gamma_{out}$	$\Gamma^l \equiv \Gamma_{und}$
Personalized PageRank	$r = 0.4$	$r = 0.4$
iMF	$k = 250, \alpha = 40, \lambda = 150$	$k = 290, \alpha = 40, \lambda = 150$
User-based kNN	$k = 120$	$k = 90$
Item-based kNN	$k = 300$	$k = 290$

Table 4. Effectiveness of the IR model adaptations and baselines. Cell color goes from red (lower) to blue (higher values) for each metric/dataset, with the top value highlighted in bold. The differences between BM25 (the best IR model) and iMF (the best baseline) are always statistically significant (two-tailed paired t-test at $p = 0.05$) except in R@10 on 200 tweets.

	1 month			200 tweets		
	P@10	R@10	nDCG@10	P@10	R@10	nDCG@10
BM25	0.0691	0.1010	0.1030	**0.0572**	0.1313	**0.1102**
BIR	0.0675	0.0943	0.0995	0.0534	0.1234	0.1016
QLL	0.0609	0.0798	0.0869	0.0490	0.1108	0.0929
QLJM	0.0580	0.0758	0.0823	0.0492	0.1124	0.0943
QLD	0.0441	0.0644	0.0682	0.0482	0.1112	0.0931
VSM	0.0191	0.0287	0.0292	0.0268	0.0597	0.0498
Money	0.0772	0.1325	0.1315	0.0476	0.1180	0.0932
Adamic-Adar	0.0676	0.0936	0.0991	0.0532	0.1236	0.1006
MCN	0.0631	0.0847	0.0920	0.0501	0.1141	0.0948
PageRank Pers.	0.0598	0.1076	0.0996	0.0336	0.0855	0.0635
Jaccard	0.0226	0.0281	0.0320	0.0304	0.0700	0.0586
iMF	**0.0834**	**0.1414**	**0.1384**	0.0541	**0.1351**	0.1045
User-based kNN	0.0805	0.1308	0.1360	0.0479	0.1211	0.0955
Item-based kNN	0.0739	0.1119	0.1174	0.0360	0.0859	0.0724
Popularity	0.0255	0.0368	0.0376	0.0225	0.0505	0.0422
Random	0.0009	0.0017	0.0013	0.0003	0.0006	0.0003

approaches, and near to the popularity baseline. Query likelihood, personalized PageRank and MCN stand as mid-packers in both datasets. Finally, classic collaborative filtering and Money show very different behaviors in both datasets: on 1 month they are among the top 5 algorithms, while on 200 tweets they are far from the best, leveled with query likelihood.
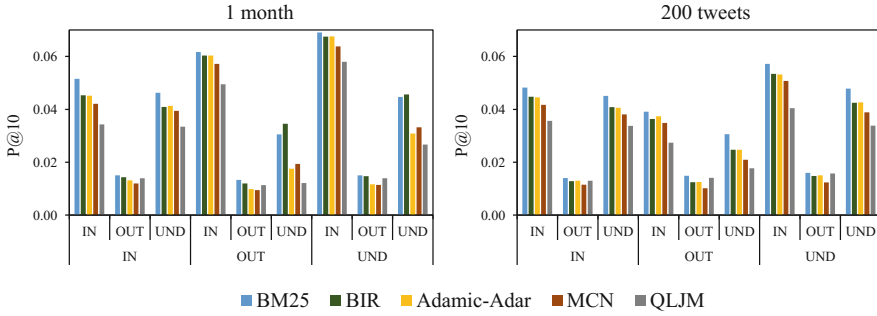
Fig. 3. P@10 values for the different possible choices for Γ^d and Γ^q on a selection of the most effective algorithms in the comparative included in Table 4.

We can also examine which neighbor orientation works best in the neighborhood-based algorithms –whether users are better represented by their followers, their followees, or both. Figure 3 shows a detailed comparison of all combinations for this setting. The outer labels on the *x* axis show the neighborhood orientation for the target user, and the inner ones for the candidate user. We can see that the undirected neighborhood Γ_{und} is consistently the most effective representation for target users, whereas the incoming neighborhood Γ_{in} works best for candidate users.

All in all, we find that BM25 makes for a highly competitive contact recommendation approach. One of the reasons for this is likely its ability to take advantage of interaction frequency (edge weights) better than any other algorithm –in fact, all other algorithms except VSM produce worse results when using a non-binary edge representation. BM25 is however not the top algorithm, since iMF overall has a slight advantage in effectiveness. Money and kNN get decent results in one dataset, but quite suboptimal in the other. We may therefore say BM25 is a decent second best in recommendation accuracy after matrix factorization. We find however important advantages to BM25 in terms of computational cost and simplicity, as we examine in the next section.

7   Complexity Analysis: BM25 Vs. Matrix Factorization

Computational cost and simplicity are critical in a commercial deployment of recommendation algorithms, which have to provide recommendations in real time. We focus on two aspects in our analysis: (a) generating recommendations from scratch, and (b) updating or retraining the algorithms each time a new user or a new link is added to the network. We first examine the cost analytically, and then we run a small test to observe the empirical difference.

7.1 Theoretical Analysis

The complexity analysis for generating recommendations for scratch is shown in Table 5, for the algorithms tested in the previous section. We can see that, in general, IR models are the fastest, along with MCN, Jaccard and Adamic-Adar, whereas implicit MF is among the costliest algorithms.

The reason why IR models (and, similarly, MCN, Jaccard and Adamic-Adar) are so fast is that we can take advantage of IR index-based optimizations, such as the "term-at-a-time" or "document-at-a-time" algorithms for fast query response-time [8]. If we store the network as an inverted index, as shown in Fig. 2, it suffices to run over the "posting lists" of target user neighbors (the "query terms") in linear time to generate a recommendation. The resulting average complexity of this is the square of the average network degree. The training time $O(|E|)$ in the table for these algorithms just corresponds to the straightforward computation of certain values such as the length of the neighborhoods.

Table 5. Running time complexity of the different algorithms, grouped by families. We show the complexity for both the full training, and the recommendation score computation (excluding the additional $\log N$ for final rank sorting). The variable m denotes the average network degree; c is the number of iterations for personalized PageRank, Money and iMF; and k represents the number of latent factors in iMF, and the number of neighbors in kNN.

Algorithms	Training	Recommendation
IR models	$O(|E|)$	$O(|\mathcal{U}|m^2)$
Jaccard, Adamic-Adar	$O(|E|)$	$O(|\mathcal{U}|m^2)$
MCN	-	$O(|\mathcal{U}|m^2)$
Random walks	$O\left(c|\mathcal{U}|^2 + c|\mathcal{U}||E|\right)$	$O(|\mathcal{U}|)$
User/Item-based kNN	$O(|E| + |\mathcal{U}|m^2\log k)$	$O\left(|\mathcal{U}|^2 k\right)$
iMF	$O(c(k^2|E| + k^3|\mathcal{U}|))$	$O\left(|\mathcal{U}|^2 k\right)$
Popularity	$O(|E|)$	$O(|\mathcal{U}|)$
Random	-	$O\left(|\mathcal{U}|^2\right)$

Implicit MF, on its side, is quadratic on the number of users, linearly multiplied by the number of latent factors. Yet worse, the same cost is incurred to produce recommendations after the training phase. Adding to this, iMF has three parameters to configure while BM25 has only two, which implies additional savings on the side of BM25 in parameter tuning cost. In terms of memory spending, assuming an all-in-memory implementation, iMF uses $2k^2|\mathcal{U}|$ decimal values, whereas BM25 only needs $3|\mathcal{U}|$ values (neighborhood length, size, and RSJ), which can make a considerable difference.

Matrix factorization is moreover not particularly flexible to incremental updates for incoming data. Update approaches have been proposed [34] by which a new link can be added in $O(mk^2 + k^3)$ time –though this does not work as an exact retraining, as it comes at the expense of incremental accuracy losses in the updated model. In contrast, BM25 can be updated in $O(1)$ for a single new link, by storing neighborhood lengths and RSJ values in the index. When a new user comes in, all values of RSJ need updating, involving an additional $O(|\mathcal{U}|)$. BM25 therefore enables fast updates, and better yet, equivalent to a full retraining. User-based kNN also enables lossless updates, but these take $O((|\mathcal{U}| + m)\log|\mathcal{U}|)$ time, which is even significantly heavier than the iMF update.

7.2   Empirical Observation

In order to observe what the theoretical analysis translates to in quantitative terms, we carry out an incremental update experiment where we test the running times for BM25, implicit MF, and –as a third-best algorithm– user-based kNN. For the 1 month network, we randomly sample 10% of the users, along with all links between them, and take this small graph as the starting point for a growing network. Over that reduced network, we train and run both recommendation algorithms. Then, we randomly sample and add one user at a time from the remaining 90% of users. For each new user, we add all its edges pointing to or from the users in the growing network. Then, we generate recommendations for all users in the subset. We continue the process until all users have been added to the growing network. We compute separately the time taken to update the recommender, and the time spent in generating the corresponding recommendations.

Figure 4 shows the time cost for both tasks: the advantage of BM25 over iMF and kNN is apparent. In incremental update (Fig. 4 right), the difference is in fact overwhelming –notice the logarithmic scale in the y axis, which means that updating BM25 is indeed orders of magnitudes faster than its two counterparts. It should moreover be noted that iMF and kNN are configured here with $k = 10$ (factors and neighbors, respectively). If we increased this parameter –as in the optimal configurations shown in Table 3– the cost would increase even further and faster.
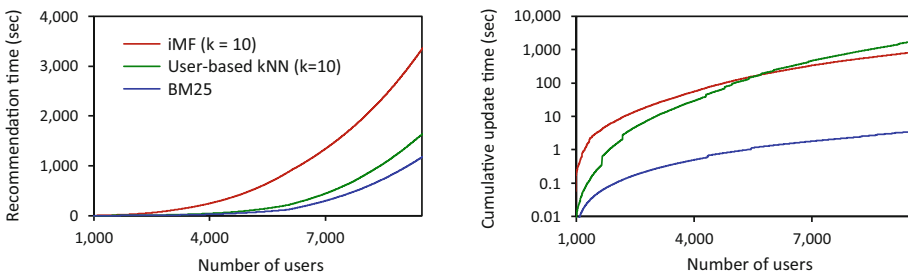

Fig. 4. Time comparison between BM25, user-based kNN and implicit matrix factorization.

8 Conclusions and Future Work

Though separately developed to much extent by different communities, text-based search and recommendation are very related tasks. This relation has been explored in prior research on the general perspective of adapting IR techniques to item recommendation [7, 31]. In our present work, we particularize this step to the recommendation of contacts in social networks. Our research has found that adapting IR models leads to empirically effective solutions, and to some extent simpler than previously developed adaptations for the general item recommendation task. We find that BM25 in particular is competitive with the best state-of-the-art approaches in terms of effectiveness over Twitter interaction networks. At the same time, IR models are orders of magnitude faster to run and update than the most effective recommendation algorithms.

Compared with alternative heuristic solutions, translating new and principled IR models to contact recommendation can add new and deeper insights to our understanding of the task and how we solve it, by importing the theory and foundations upon which the IR models were developed. Reciprocally, this can contribute to a broader perspective on IR models, their meaning, interpretation, and usefulness in different tasks, bringing higher levels of abstraction. We have thus found for instance that IR models tend to take better advantage of user-user interaction frequency than heuristic algorithms. We have likewise observed that followers seem to describe the social value of candidate recommendations better than followees, whereas the union of both consistently appears to best represent the social needs of target users.

We envision continuing this line of research to deeper levels in future work. We also plan to extend our current research by considering further evaluation dimensions beyond accuracy, such as recommendation novelty and diversity [9, 29], or the effects that recommendation can have on the evolution of the network structure [3, 28].

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Soc. Netw. **25**(3), 211–230 (2003)
2. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
3. Aiello, L., Barbieri, N.: Evolution of ego-networks in social media with link recommendations. In: 10th ACM International Conference on Web Search and Data Mining (WSDM 2017), pp. 111–120. ACM, New York (2017)
4. Backstrom, L., Leskovec, J.: Supervised random walks: predicting and recommending links in social networks. In: 4th ACM International Conference on Web Search and Data Mining (WSDM 2011), pp. 635–644. ACM, New York (2011)
5. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval: the concepts and technology behind search, 2nd edn. Addison-Wesley, Harlow (2011)

6. Belkin, N.J., Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? Commun. ACM **35**(12), 29–38 (1992)
7. Bellogín, A., Wang, J., Castells, P.: Bridging memory-based collaborative filtering and text retrieval. Inf. Retrieval **16**(6), 697–724 (2013)
8. Büttcher, S., Clarke, C.L.A., Cormack, G.V.: Information retrieval: implementing and evaluating search engines. MIT Press, Cambridge (2010)
9. Castells, P., Hurley, N.J., Vargas, S.: Novelty and Diversity in Recommender Systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 881–918. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_26
10. Goel, A., Gupta, P., Sirois, J., Wang, D., Sharma, A., Gurumurthy, S.: The who-to-follow system at twitter: strategy, algorithms and revenue impact. Interfaces **45**(1), 98–107 (2015)
11. Goel, A., Gupta, P., Sirois, J., Wang, D., Sharma, A., Gurumurthy, S.: WTF: the who to follow service at Twitter. In: 22nd International Conference on World Wide Web (WWW 2013), pp. 505–514. ACM, New York (2013)
12. Guy, I.: Social Recommender Systems. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, 2nd edn, pp. 511–543. Springer, New York (2015)
13. Hannon, J., Bennet, M., Smyth, B.: Recommending Twitter users to follow using content and collaborative filtering approaches. In: 4th ACM conference on Recommender Systems (RecSys 2010), pp. 199–206. ACM, New York (2010)
14. Al Hasan, M., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: SDM 06 Workshop on Link Analysis, Counterterrorism and Security at the 5th SIAM International Conference of Data Mining (2006)
15. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: 8th IEEE International Conference on Data Mining (ICDM 2008), pp. 263–272, IEEE Press, New York (2008)
16. Jaccard, P.: Étude de la distribution florale dans une portion des Alpes et du Jura. Bulletin de la Société Vaudoise des Sciences Naturelles **37**(142), 547–579 (1901)
17. Jelinek, F., Mercer, R.L.: Interpolated estimation of Markov source parameters from sparse data. In: Gelsema, E.S., Kanal, L.N. (eds.) Pattern Recognition in Practice, pp. 381–397. North-Holland, Amsterdam (1980)
18. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inform. Sci. Technol. **58**(7), 1019–1031 (2007)
19. Lü, L., Zhou, T.: Link prediction in social networks: a survey. Phys. A **390**(6), 1150–1170 (2010)
20. Mackay, D., Bauman, L.: A hierarchical Dirichlet language model. Nat. Lang. Eng. **1**(3), 289–307 (1995)
21. Ning, X., Desrosiers, C., Karypis, G.: A comprehensive survey of neighborhood-based recommendation methods. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 37–76. Springer, Boston, MA (2015). https://doi.org/10.1007/978-1-4899-7637-6_2
22. Ponte, J.M., Croft, W.B.: A language modelling approach to information retrieval. In: 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998), pp. 275–281. ACM Press, New York (1998)
23. RankSys library. http://ranksys.org. Accessed 11 Oct 2018
24. Robertson, S.E.: The probability ranking principle in IR. J. Doc. **33**(4), 294–304 (1977)
25. Robertson, S.E., Zaragoza, H.: The probabilistic relevance framework: BM25 and beyond. Found. Trends Inform. Retrieval **3**(4), 333–389 (2009)
26. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)

27. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
28. Sanz-Cruzado, J., Castells, P.: Enhancing structural diversity in social networks by recommending weak ties. In: 12th ACM Conference on Recommender Systems (RecSys 2018), pp. 233–241. ACM, New York (2018)
29. Sanz-Cruzado, J., Pepa, S.M., Castells, P.: Structural novelty and diversity in link prediction. In: 9th International Workshop on Modelling Social Media (MSM 2018) at The Web Conference (WWW Companion 2018), pp. 1347–1351. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland (2018)
30. Sanz-Cruzado, J., Castells, P.: Contact recommendations in social networks. In: Berkovsky, S., Cantador, I., Tikk, D. (eds.) Collaborative recommendations: algorithms, Practical Challenges and Applications, pp. 525–577. World Scientific Publishing, Singapore (2018)
31. Valcarce, D.: Exploring statistical language models for recommender systems. In: 9th ACM Conference on Recommender Systems (RecSys 2015), pp. 375–378. ACM Press, New York (2015)
32. Valcarce, D., Parapar, J., Barreiro, A.: Axiomatic analysis of language modelling of recommender systems. Int. J. Uncertainty, Fuzziness Knowl.-based Syst. **25**(Suppl. 2), 113–127 (2017)
33. White, S., Smyth, P.: Algorithms for estimating relative importance in networks. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003), pp. 266–275. ACM, New York (2003)
34. Yu, T., Mengshoel, O., Jude, A., Feller, E., Forgeat, J., Radia, N.: Incremental learning for matrix factorization in recommender systems. In: 2016 IEEE Conference on Big Data (BigData 2016), pp. 1056–1063. IEEE Press, New York (2016)